# **Image Classification Game**

This **Snap! game** uses **Nvidia Jetson** capability to classify images.

#### Offline Snap! downloading

Please download and open Offline version of Snap! for our project. Go to https://snap.berkeley.edu/offline and follow the steps.

# Snap! files' downloading

Please open the link Classification Game to download our project on your computer. Probably you would see the xml in raw format. Click the right button of your mouse and save it on the disk.



# Web camera Image in Snap!

You can get picture from your web camera in Snap!.

video capture block to enable video capturing.



• Change value of **set video transparency** block to 0 for clear image.

```
set video transparency to 0
```

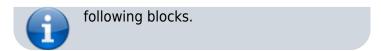
• video snap on stage block reports picture from stage.



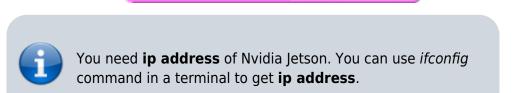
### **Connection to Jetson from Snap!**

If you have not imported it yet, please download jetson blocks and import it to your Snap! project.





• Use **connect to Jetson url** block to connect Jetson.



connect to jetson url ws://ip\_address:4040

• Store the value of **jetson\_name** in a variable.



• Store the value of **connect to Jetson url** block in a variable for later use.

```
et jetson to connect to jetson url join ws:// (jetson_name
```

## **Response from classification**

Here we will send video snap on stage to Jetson for processing. Jetson will respond back class name, confidence value and class ID.



- Use get response from Jetson block to send image , and get class name and confidence value.
  - First input slot is for **jetson** variable that stores websocket data.
  - Second input slot is for **costume** you want to be classified by Nvidia Jetson.

```
set response to send picture to jetson jetson video snap on Stap
```

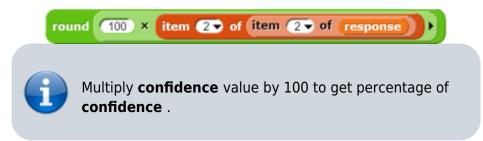
### Class name and confidence value

This section will demonstrate how to handle response variable to access class name and confidence value.

• class name is the 2nd item of 1st item of response block.



• confidence value is 2nd item of 2nd item of response block.





You can create custom blocks, to get **class name** *get class* name from response and to get **confidence** *get confidence* from response .

#### **Speech functionality**

**Speech functionality** is available as a library in Snap!. Select *export libraries* from settings then choose *speech module* .

• Use join block to create text like I am confidence percent sure, it is class name.

```
join round 100 × item 2→ of item 2→ of response | | |
percent-sure vittis | item 2→ of item 1→ of response | |
```

Use speak and wait block to read text a loud.

```
| round | 100 × | item | 2 of | item | 2 of | response | 100 | response |
```

#### Repeat block for game

Last step is adding loop for the game.

• Use repeat block and put script inside of it.

```
set response to send picture to jetson jetson video snap on Stage speak

speak

round 100 × item 2 of item 2 of response

join
percentsure that it is item 2 of item 1 of response

with en-GB accent
and pitch 1 rate 1

wait 3 secs
```

14:38



This example used **repeat until** block to break loop when space key pressed.

You can download full game from Github page of EOLab-

https://student-wiki.eolab.de/ - HSRW EOLab Students Wiki

Permanent link:

https://student-wiki.eolab.de/doku.php?id=snapcon2022:image-classification-game&rev=1659786975



